IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received October 22, 2017, accepted November 23, 2017, date of publication December 8, 2017,
date of current version February 14, 2018.

Digital Object Identifier 10.1109/ACCESS.2017.2781280

Line Junction Detection Without Prior-Delineation of Curvilinear Structure in Biomedical Images

HANJIN ZHANG[1], YANG YANG[2], Member, IEEE, and HONGBIN SHEN[1]

[1]Institute of Image Processing and Pattern Recognition, Shanghai Jiao Tong University, and Key Laboratory of System Control and Information Processing, Ministry of Education of China, Shanghai, 200240, China

[2]Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, China

Corresponding author: Hongbin Shen (hbshen@sjtu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61671288, Grant 91530321, Grant 61603161, and Grant 61725302, and in part by the Science and Technology Commission of Shanghai Municipality under Grant 16JC1404300, Grant 17JC1403500, and Grant 16ZR1448700.

ABSTRACT The line junction detection is a fundamental step in many computer vision applications, especially in biomedical image analysis. Most of the existing studies determine the junction position after delineating curvilinear structure, thus the detection accuracy relies heavily on the previous steps for curvilinear extraction, such as image segmentation and skeletonization. In this paper, we treat the detection of line junctions as an independent task without prior knowledge of curvilinear structures. We present the mathematical definition and properties of line junctions, and propose a new method called Junction Recognition (JUNR). It first maps the raw images into score matrices (or called score images) by the measurements based on line junction properties, then detects and screens blobs from the score images for identifying the regions covering junction points. Finally, it refines the locations of line junctions as well as their branch properties. A distinct advantage of JUNR is that it can be directly applied to raw images without knowing curvilinear structure beforehand. Besides, since JUNR is a rule-based method, it requires no training data and avoids the labor-intensive labeling work. We conducted experiments on two typical kinds of biomedical images, including both simulated and real images with curvilinear structures. Both qualitative and quantitative results demonstrate its good performance for junction detection and characterization.

INDEX TERMS Biomedical image processing, blob screening, line junction detection.

I. INTRODUCTION

The extraction of line junctions in images is a crucial task in many complex computer vision applications, especially in biomedical image analysis. For instance, in neuron images, the detection and localization of critical points, e.g., junctions and terminations, is a key step in many neuron reconstruction studies [1]; in retinal fundus images, the vascular junction information can assist in diagnosing complex diseases, such as diabetic retinopathy and angiomatosis [2]–[4].

Generally, junctions fall into two categories, i.e., general junctions and line junctions. A general junction is the place where two or more sector regions meet, which is often simply regarded as corner in some literatures [5]–[8]; while a line junction is usually considered as the line bifurcation which has two or more distinct branches, or the intersection point where two or more lines crossover each other. To avoid ambiguity, in this paper, we call general junctions that are not line junctions as natural junctions. Fig. 1 shows some examples of natural junctions and line junctions. According to the number of branches, both the two kinds of junctions can be categorized into L-, Y- (or T-), X-junctions and even higher-order junctions. In most real images, the L-line junctions have little significance, thus we do not distinguish them from high-curvature line points.

Natural junction is one of the most basic elements in nearly all kinds of images. A lot of approaches for the identification of natural junctions have been proposed, which have laid theoretical foundation for the development of line junction detection methods. These approaches can be divided into three major types, which are based on signal-intensity, contour and template, respectively. The signal-intensity based methods detect changes of local signal intensity in the images. They mostly function as a corner detector, i.e., searching the corner region which has locally peak value with respect

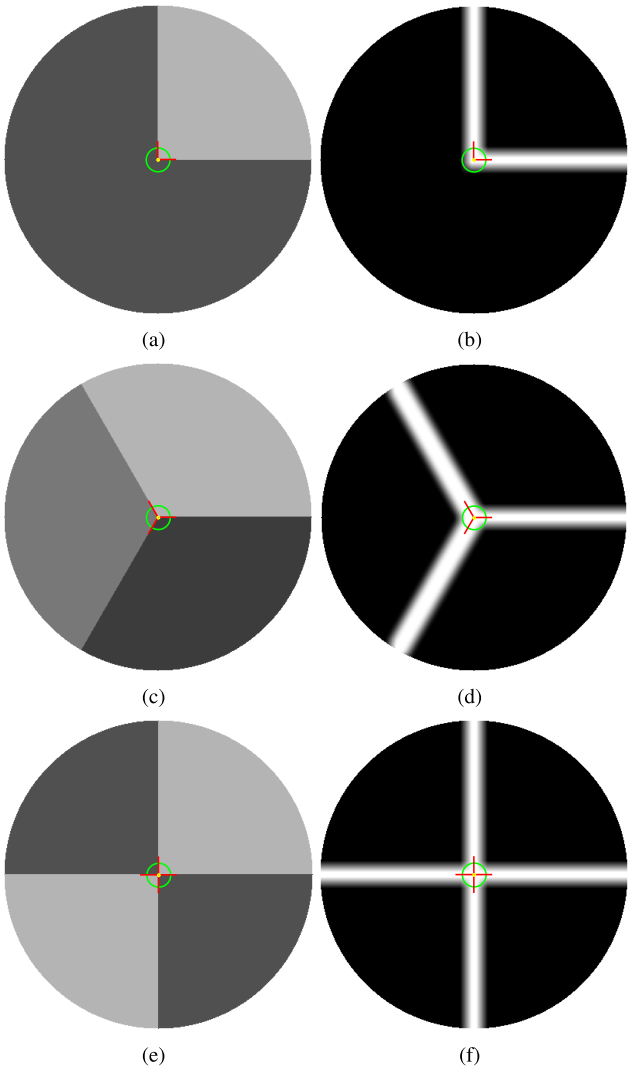


FIGURE 1. Examples of different junction types. (a), (c) and (f) are L-, Y-, X-natural junctions, and (b), (d) and (f) are L-, Y-, X-line junctions.

to a certain criterion. Representative works include Harris's method [9], SUSAN (smallest univalue segment assimilating nucleus) [10], the FAST (Features from Accelerated Segment Test) corner detector [11], [12], etc. Contour based methods basically aim to estimate contours and then localize the junctions, such as Maire's method [13], JUDOCA (JUnction Detection Operator Based on Circumferential Anchors) [8], and Xia's contrary detection theory [6]. Template based methods construct template models to represent the junction structures and perform template matching [5], [7]. And the natural junction detection is widely used in some complex visual applications, such as dental identification [14] and key point extraction in RGB-D images [15].

Different from detecting natural junctions, the identification of line junctions is often regarded as a post-processing step of curvilinear structure analysis [16]–[20], thus its performance relies heavily on the quality of image segmentation, skeletonization and other previous steps for line recognition (as shown in Fig. 2). A representative and general-purpose method was proposed by Deschenes *et al.* [21], [22], in which line junctions and terminations are identified based on the estimation of line curvature. Azzopardi and Petkov [18], [23] presented a new filter called combination of shifted filter responses (COSFIRE).

Actually, in many real applications, identification of the curvilinear structure from background is not an easy job. Thus, the methods of line junction detection directly working on raw images are relatively few. Chen et al.'s method is an example [24], they extended 2-D image into a 3-D space with the third element denoting the orientations. This approach effectively captures junction region and specifies junction branches, but it is unable to provide the precise location of junction centers. Su et al.'s method [25], [26] also shows good performance and wide applications on different images, but their quantitative results seem not sufficient. Besides, a lot of methods were designed for certain types of images, like binary images, or for specific applications, such as the identification of line-drawing junctions [27], hand-written characters [28] and vessel or nervus [16], [29]–[31].

Considering that independent methods for line junction detection have less restriction and more flexibility, we focus on exploring new robust and versatile methods to deal with complex junction structures. Here, we propose a new line junction detection protocol called JUNR (JUNction Recognition), which directly works on raw images and needs no curvilinear structure knowledge. JUNR aims to identify the line junction centers and also obtain their properties, such as the number of junction branches and their directions. First, it constructs a measurement to evaluate the potential to be a line junction center for each point, thus generating a score matrix for each image. Then, it considers the matrix as a special image, and exploits blob detection to identify the regions where the junction centers locate. By screening for the most possible blobs which cover line junction centers, JUNR determines the line junction center location via a combined measurement. After that, JUNR specifies junction branch characteristics through finding adjacent filter responses. Our experimental results show that the new method has good performance on various images, including synthesized neuronal and real retinal fundus images. In a quantitative analysis on both synthesized and real biomedical datasets, it achieves better or comparable performance to the state of art methods specialized for those datasets.

The rest of this paper is organized as follows: Section 2 introduces the line junction definition and summarizes several important properties; Section 3 describes the proposed algorithm JUNR in details; Section 4 presents the qualitative and quantitative experimental results of JUNR, and compares it with the state-of-the-art methods for line junction detection; and the last section concludes the paper.

II. LINE JUNCTION DEFINITION AND PROPERTIES

In this paper, we treat the detection of line junction as an independent task rather than an add-on procedure after the identification of curvilinear structure. Therefore, it is

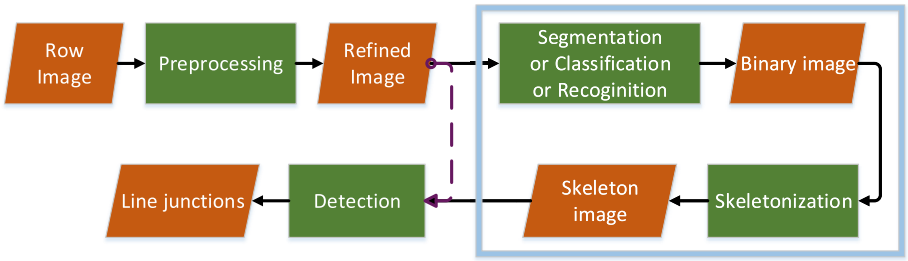


FIGURE 2. Typical flowchart of line junction detection.

necessary to use formal definitions and mathematical properties to describe the task. Inspired by the definitions of natural junctions [6], [7], here we give the formal definitions of line junction and branch line.

Definition 1 (Line Junction): In an image which contains complex linear or curvilinear structures, a junction is the region which locates at the line bifurcation or line intersection, and consists of four elements as shown in Eq. ((1)),

$$J : \{c, r, \theta_{m=1}^{M}, L(\theta_m)\}, \tag{1}$$

where c is the junction center point with the coordinate (x, y), r denotes the radius of junction region centered at c, $\theta_{m=1}^{M}$ is the junction branch set with M elements, and $L(\theta_m)$ denotes the line with the dominant orientation of the m-th branch.

Definition 2 (Branch Line): A branch line L is a local linear structure adjacent to a junction. It can be represented by four elements,

$$L(\theta_m) : \{c(\theta_m), p(s), w, l\}, \tag{2}$$

where $c(\theta_m)$ denotes the centerline of the line with the dominant orientation of θ_m, $p(s)$ represents the profile of the line structure, w is the effective width of the profile, and the last term l is the length of line segment in local region, supposing that the line in local region is straight. For a general gray image, the line profile is usually approximated as a Gaussian curve or parabola curve [24], [32], [33]. In this paper, we adopts the Gaussian model as defined in Eq. ((3)),

$$p(s) = \begin{cases} \dfrac{1}{\sqrt{2\pi}\sigma} exp(-\dfrac{(s-\mu)^2}{2\sigma^2}) & -\dfrac{w}{2} \leqslant s \leqslant \dfrac{w}{2} \\ 0 & \text{others,} \end{cases} \tag{3}$$

where μ and σ are the mean and standard deviation of the Gaussian curve, respectively. To limit the width of the line profile, we narrow the original Gaussian curve in the range of $[-\frac{w}{2}, \frac{w}{2}]$, where $w \leq 2\sigma$.

Given these definitions, we introduce three properties of line junctions as follows. Here we consider an ideal image which contains the curvilinear structure as defined in Eq. ((3)), and the background is homogeneous with low noise. Without loss of generality, we just consider bright lines in the dark background, and it is easy to extend to the opposite situation. Regardless of the effect of edge or blob, points in an image can be divided into three subsets, the line junction point set S_J, the curvilinear structure point set S_L, and the background point set S_B.

Property 1 (The Intensity Distribution Property): Assume that a set of concentric rings are centered at an arbitrary point p, the intensity distribution of the points within the rings varies with the position of p. If p is a line junction center (i.e., $p \in S_J$), as the radius r increases, the intensity distribution of the points in the annulus (centered at p, with the inner radius r and outer radius $r + dr$) will first change dramatically (the mean of intensity μ_I decreases and the variance σ_I increases), and then become stable. If $p \in S_L$, as r increases, the intensity distribution has the same change tendency. The only difference with junction points is that the number of peak branches is less than 3. If $p \in S_B$, as r increases, the intensity has no obvious change, and intensity distribution remains nearly flat with no significant peak.

Property 2 (The Orientation Flux Property): The definition of orientation flux was proposed in [34]. For a point p in image I with local circle region C_r, the oriented flux along direction $\boldsymbol{v}$ is computed by the image gradient projection, i.e.,

$$f(p, \boldsymbol{v}; r) = \int_{\partial C_r} (\nabla(G * I(p + \boldsymbol{h}) \cdot \boldsymbol{v})\boldsymbol{v}) \cdot \frac{\boldsymbol{h}}{|\boldsymbol{h}|} da, \tag{4}$$

where $\boldsymbol{h}$ denotes the position normal vector of C_r, da is the infinitesimal length on ∂C_r, r is the radius of local inclosed circle region, and G is a Gaussian function with a scale factor of an individual pixel. It is easy to calculate the flux $f(p, v; r)$ using a convolution by the divergence theorem,

$$\begin{aligned} f(p, \boldsymbol{v}; r) &= \boldsymbol{v}^T \{(\psi_p G) * b_{C_r} * I\}\boldsymbol{v} \\ &= \boldsymbol{v}^T \boldsymbol{Q}_{p,r} \boldsymbol{v}, \end{aligned} \tag{5}$$

where $\psi_{\boldsymbol{p}} G$ is the second derivatives of Gaussian function G and b_{C_r} is the indicator function of the local circle, and we use a matrix $\boldsymbol{Q}_{p,r}$ to denote $\{(\psi_p G) * b_{C_r} * I\}$. Here, $f(p, v; r)$ should be minimized to find the line direction and estimate the local line width. And this minimization problem can be transformed into a generalized eigenvalue computation problem of the matrix $\boldsymbol{Q}_{p,r}$ [34]. Suppose that $\lambda_1(p, r)$, $\lambda_2(p, r)$ are the two eigenvalues of $\boldsymbol{Q}_{p,r}$, and $\lambda_1(p, r)$ is the bigger one. If $p \in S_L$, then the outward flux is dominated along the curvilinear direction at this point, thus $|\lambda_1(p, r)|$ would be much larger than $|\lambda_2(p, r)|$; if $p \in S_B$, since it has no distinct difference with adjacent points, the outward flux is similar in

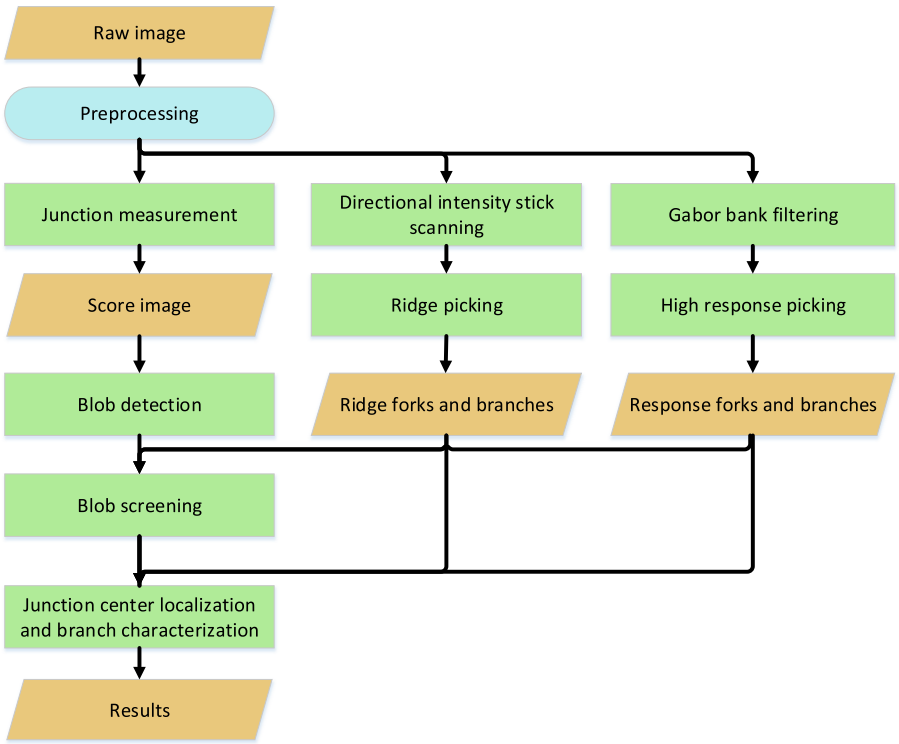


FIGURE 3. The flowchart of JUNR.

every direction, i.e., $\lambda_1(p, r) \approx \lambda_2(p, r) \approx 0$; if $p \in S_J$, it is hard to estimate the magnitude of outward flux, because the junction has several dominant directions along the branches.

Property 3 (The Direction Filter Response Property): When an image is convolved with a directional filter, the responses for the junction, the line structure and the background are different. With an appropriate scale, considering a local region $I_{\theta_I}(x, y)$ filtered with a directional filter $F_\theta(x, y)$, the response can be formulated as,

$$R_\theta(x, y; \theta_I) = I_{\theta_I}(x, y) * F_\theta(x, y). \tag{6}$$

Let the valid angle range of the filter be $[\theta - \varsigma, \theta + \varsigma]$. According to [24], for a typical line structure, the response follows the properties below:

- If the line direction is equal to the filter direction, i.e., $\theta_L = \theta$, then $R_\theta(x, y; \theta_L) = max(L_{\theta_L}(x, y) * F_\theta(x, y))$;
- The response of the line direction in the valid angle range is much larger than that out of the range, i.e., $R_\theta(x, y; \theta_{L1}) \gg R_\theta(x, y; \theta_{L2})$ where $\theta_{L1} \in [\theta - \varsigma, \theta + \varsigma]$ and $\theta_{L2} \notin [\theta - \varsigma, \theta + \varsigma]$;
- In the valid angle range, the responses of two lines are equal if they are symmetrical to the dominant direction θ, i.e., $R_\theta(x, y; \theta_{L1}) = R_\theta(x, y; \theta_{L2})$, where $\theta_{L1} - \theta = \theta - \theta_{L2}$ and $\theta_{L1}, \theta_{L2} \in [\theta - \varsigma, \theta + \varsigma]$.

For a typical line region, the response of the whole filter bank has distinct peak value along the line direction. For a junction region, the response of the whole filter bank is multi-modality, because it has several branches and obtains peak responses along branch directions. For a homogeneous background region, no direction has notable difference.

III. THE JUNR ALGORITHM

A. METHODOLOGY OVERVIEW

The proposed method, JUNR consists of four steps. In the first step, we construct a measurement for each point in the image to assess its potential as a line junction. Accordingly, the original image is converted into a score matrix, where each element in the matrix is the score of a point in the image. For the second step, by regarding the score matrix as a special image, where the scores are pixel intensities, we exploit a blob detection method to identify the regions which may contain line junctions. In the third step, we screen the blobs and keep the blobs with high confidence of covering line junctions. And in the last step, we locate the junction center and determine the number and orientations of junction branches. For real images, some preprocessing operations are needed, such as background homogenization and denoising. The flowchart of the proposed method is depicted in Fig. 3.

B. THE MEASUREMENT FOR EVALUATING LINE JUNCTION CENTERS

In order to detect line junctions, a measurement is needed to evaluate the potential of being a junction center for each point in the image. Based on the aforementioned properties, we define a comprehensive measurement, as shown in Eq. ((7)),

$$\phi(p) = \sum_{i=1}^{K} \alpha_i M_i, \tag{7}$$

where $\phi(p)$ is the score of point p, K is equal to 3, α_i is the coefficient for the i-th measurement, and M_is denote the measurements for the effect of intensity distribution, optimal

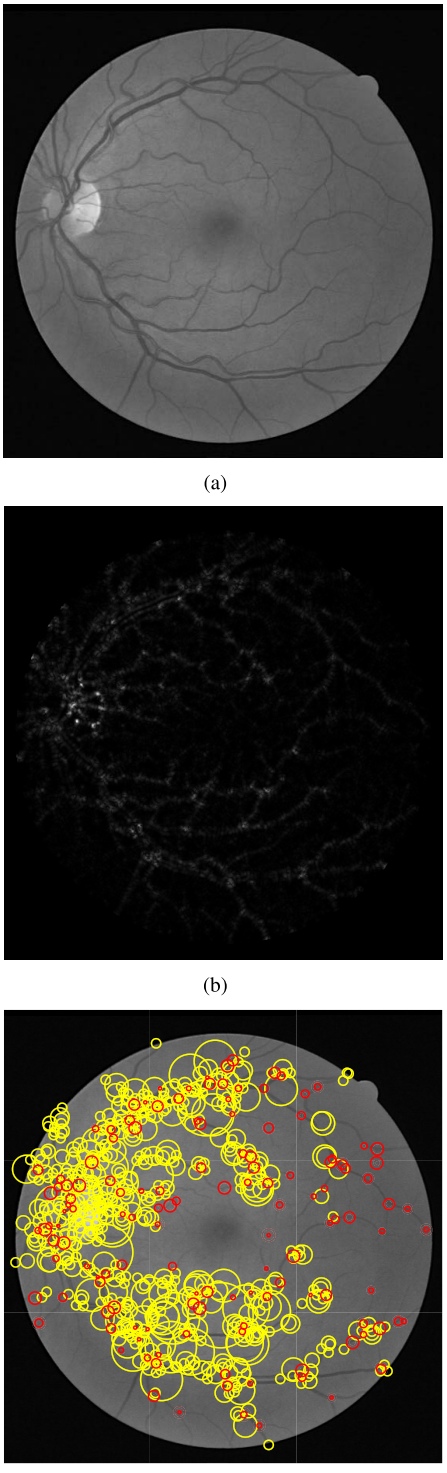


FIGURE 4. The extraction and screening of junction blobs. (a) is the raw gray retinal vessel image, (b) is the normalized score image, and (c) indicates the blob detection and selection. Both yellow and red circles denote extracted blobs. Yellow ones are filtered while red ones are remained.

orientation flux (OOF) and their combined effect, respectively. The M_is are defined as follows.

- $M_1 = norm(\lambda_{min})$, the normalized minimal eigenvalue of correlation matrix $A = \frac{1}{N}\sum_{j=1}^{N}((g_j) * (g_j)^T)$, where g_j denotes the gradient of each element in this matrix. This measurement is the same as the minimal eigenvalue measurement for corner detection [35]. The principle is that line junction and corner have a common property, i.e., the variations of the local autocorrelation matrix for the gradients along all directions are great. Note that line junction center may overlap with corner point when the line width is sufficiently narrow, thus before constructing the correlation matrix, we use a Gaussian kernel with different variances to convolve with the image window for estimating the effect of line width.
- $M_2 = norm(\lambda_{min}^{OOF})$, the normalized minimal eigenvalue of orientation flux matrix. Optimal orientation flux (OOF) computes the optimal projection orientation by minimizing the inward oriented flux at the boundary of localized circles of different scales. And this optimizing problem is transformed into a generalized eigenvalue computation problem. According to Property 2 in Section II, the magnitude of eigenvalues varies with the position of point. For junction center points, the eigenvalues are much larger than those of line points or background points, thus the minimal eigenvalue is used as the measurement for junction points.
- $M_3 = R_C^2/R_O$, where R_C is the ratio of the two eigenvalues from the correlation matrix, and R_O denotes the ratio of the two eigenvalues from OOF matrix. This measure considers the combined effect of the first two measurements. A junction point is expected to have larger R_C and smaller R_O than non-junction points.

The final score is a linear combination of the three measurements. As mentioned in Section 2, a line junction has similar properties as a natural junction or a corner, and the impact of the first measurement should be much larger than other two measures. Therefore, the weights are set as $\alpha_1 \gg \alpha_2 > \alpha_3$ to strengthen the effect of corner metric. Then the whole image or field of view (FOV) is mapped into a score matrix, and a high score indicates a high possibility of being a junction center.

C. DETECTION OF CANDIDATE JUNCTION BLOBS

After assigning a score for each pixel, the original image is mapped into a score matrix, which can be regarded as a special image. We call it score image. Fig. 4(a) shows an example. Intuitively, the pixels with highest scores are junction centers. However, the per pixel prediction, i.e., predicting each pixel respectively, has some disadvantages. First, it ignores the dependencies of labels (junction point, line point, background point) for nearby pixels. Second, points with local peak values are not necessarily junction centers, because many points near the edge of intersection have even higher scores than the true junction centers. Third, the accuracy can be easily affected by outliers or noisy points.

To overcome these drawbacks, we design a detection method which considers the dependencies of nearby pixels to improve the prediction accuracy. Especially, we adopt the Laplace of Gaussian blob extraction algorithm [36] to

extract blobs in the score image, which are candidate junction regions. (The blob metric value is denoted as BM_i, where i denotes the i-th blob in the selected blob set.) To avoid missing junction centers, in this step we include sufficient blobs for further screening. Fig. 4(b) shows all the blobs extracted from the score image mapped for a DRIVE image (from the retinal vessel image dataset published by Staal *et al.* [37]).

D. SCREENING THE JUNCTION BLOBS

Given the junction blobs, how to select high-quality blobs is a key problem. The high-quality blobs should cover as many line junctions as possible and contain few false positives in the meantime. Intuitively, junction centers are located at the intersection of centerlines, i.e., all junction center points are located on centerlines. However, we do not have any prior knowledge of the centerlines.

Therefore, in this step, we use the approximate ridge lines instead. And we extract the approximate ridge points by both the intensity profile of lines [38] and the maximal directional filter response [24]. The junction blobs that are not crossed by any approximate ridge line are discarded. In order to obtain the approximate ridge line points, we adopt a base line algorithm [38] and calculate the intensity distribution of multi-scale lines along equally spaced directions in the local window centered by each pixel. To distinguish between the ''line'' in base line algorithm and ''line'' in a line structure, here we use ''stick'' instead of ''line'' to denote the former one. Suppose $I(x, y)$ is the intensity of point (x, y) which is the center of a window with size $W \times W$, the average intensity values of the window is I^W_{avg}. The initial screening step selects the points with high intensity values, with the criterion $I(x, y) > \tau_1 I^W_{avg}$ as mentioned in [25], where τ_1 denotes the ratio threshold. The base sticks (with length $L < W$) orient towards 12 directions (evenly spaced), respectively, and pass through the window center (x, y). The average intensity value along each stick is computed, and the stick which has the maximum intensity value (I^L_{max}) is the winning stick, and the stick response is calculated as,

$$I_R = I^L_{max} - I^W_{avg}. \tag{8}$$

We further screen the candidate points which satisfy $I_R > I_{th}$, where $I_{th} = \tau_2 I^W_{avg}$. The winning stick direction is approximately equal to the line direction at position (x, y). Thus the normal direction of the line profile is approximately equal to the direction which is perpendicular to the winning stick. Then the pixels along the normal direction of sticks are exploited to determine whether the center of window is located at the peak of the profile, by checking whether Eq. ((9)) holds,

$$[I(x, y) - I(x_{n^-}, y_{n^-})][I(x, y) - I(x_{n^+}, y_{n^+})] > 0, \tag{9}$$

where (x_{n^-}, y_{n^-}) and (x_{n^+}, y_{n^+}) denote the points located at the two sides of (x, y) respectively with the same distance along the normal direction of the stick. L can vary in multi-scale space, and we select multiple pairs of (x_{n^-}, y_{n^-}) and (x_{n^+}, y_{n^+}) with different distances to the center for verification.

Moreover, we also use Gabor filter bank to locate the ridge-like points. Reference [24] has demonstrated that strong response can be obtained for line and junction by tuning Gabor filters. However, Gabor filter could also produce strong response on the edge of images. Considering that the linear structure has a property that its first-order derivative is close to zero while second-order derivative is large [33], thus before convolving with the Gabor filter bank, we first take the second-order derivative of images along the horizontal and vertical coordinates respectively, then calculate their convolution with filter banks separately, i.e.,

$$R_D(x, y) = I_D(x, y) * h_{(\theta,\nu)}(x, y, \sigma, \omega), \tag{10}$$

where I_D is one of the second-order derivatives, $I_D \in \{I_{xx}, I_{xy}, I_{yy}\}$. After that, we re-normalize each derivative response and get the maximum value of each point $R_{max}(x, y) = max\{R_{xx}(x, y), R_{xy}(x, y), R_{yy}(x, y)\}$ as the final Gabor filter response. Then non-maximal suppression is used to screen points and the remaining points can be approximately regarded as approximate ridge line points. After conducting the above two methods, i.e., by using intensity profiles and Gabor filter bank convolution, we obtain two candidate sets of approximate ridge line points from these two methods respectively. Furthermore, we utilize morphology operations to fill the tiny gaps and fix discontinuous locations in both sets.

Once approximate ridge lines are extracted, the locations of forks are also known, where the line junction centers are most likely located. Therefore, the initial screening step is to select the blobs which contain a fork identified from either set mentioned above. Moreover, we eliminate the overlapped junction blobs by considering the following two cases, where D_{ij} denote the distance between the centers of the ith blob and the jth blob,

- Case I: If two blobs have a small overlap ($max\{\sigma_i, \sigma_j\} < D_{ij} \leq (\sigma_i + \sigma_j)$, where σ denotes the radius of a blob, then both of them will be kept.
- Case II: If the center of a blob locates within another blob, i.e., $D_{ij} \leq max\{\sigma_i, \sigma_j\}$, then further judgement is required.

Examples of these two cases are shown in Fig. 5.

In case II, the blob will be kept if it satisfies the following condition: the number of identified approximate ridge points is larger than the threshold T (In all our experiments $T = 2\sigma'$, where σ' denotes the minimum integer that is greater than the blob radius σ). An additional selection criterion is that, if a blob center is located in another blob, we will compare the proportion of identified approximate ridge points among the whole set of points in the blob, and select the blob i if $(n^c_i/n^m_i) > (n^c_j/n^m_j)$ where n^c_i and n^c_j denote the number of identified approximate ridge points in the ith and jth blob respectively, and n^m_i, n^m_j denote the total numbers of points in the two blobs. If $(n^c_i/n^m_i) = (n^c_j/n^m_j)$ then we keep the blob with the higher blob metric value (ν) and discard the other.

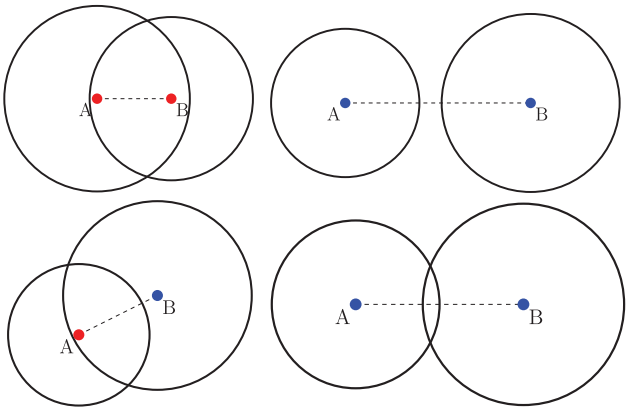


FIGURE 5. Typical overlapped blobs. The blobs centered with blue points are kept, and the blobs centered with red points need further judgement.

In Fig. 1, the blobs denoted by yellow circles are discarded by the screening step. Finally, the number of the remaining blobs is equal to the number of line junctions we will get. The pipeline of blob screening is shown in Algorithm 1. Some input notations are explained as follows. For the input blob set $S_b = \{(c_m, \sigma_m, \nu_m), m = 1, 2, \ldots, M\}$, c_m, σ_m and ν_m denote the coordinate, radius and metric value of the m-th blob, respectively. For the intensity fork set $S_{f1} = \{(c_{n1}, l_{n1}), n1 = 1, 2, \ldots, N1\}$, c_{n1} and l_{n1} denote the coordinate and branch information of the $n1$th fork, and $N1$ denote the total number of intensity forks. For the filter response fork set $S_{f2} = \{(c_{n2}, l_{n2}), n2 = 1, 2, \ldots, N2\}$, c_{n2}, l_{n2} and $N2$ can be defined similarly.

E. LINE JUNCTION CENTER LOCALIZATION AND BRANCH CHARACTERISTICS ESTIMATION

For a point in a junction blob selected by the previous steps, it can be regarded as a junction center point if the following conditions hold simultaneously. First, it is located at or close to the fork of ridge lines. Second, it is located at or close to a blob center. As we have obtained two sets of ridge line forks (See Section III-D), by using intensity information and Gabor filter bank convolution, respectively. Here we use cross points of ridge lines and converging points of branches to distinguish these two kinds of forks. Then we compute the distances from each query point J_q to the three types of points, namely the ridge line cross point, branch converging point and junction blob center, denoted by J_c, J_b and C_j, respectively. Let D_1, D_2 and D_3 denote these three distances, i.e.,

$$D_1 = min|J_q - J_c| \tag{11}$$
$$D_2 = min|J_q - J_b| \tag{12}$$
$$D_3 = min|J_q - C_j| \tag{13}$$

The final junction position is determined by Eq. ((14)),

$$J_q = argmin \sum_{i=1}^{3} \beta_i D_i, \tag{14}$$

Algorithm 1 Blob Screening Algorithm

Require: Initial blob set $S_b = \{(c_m, \sigma_m, \nu_m), m = 1, 2, \ldots, M\}$, intensity fork set $S_{f1} = \{(c_{n1}, l_{n1}), n1 = 1, 2, \ldots, N1\}$, filter response fork set $S_{f2} = \{(c_{n2}, l_{n2}), n2 = 1, 2, \ldots, N2\}$, distance tolerance ϵ, intensity ridge point set $S_{r1} = \{c_{nr1}, nr1 = 1, 2, \ldots, Nr1\}$, filter response ridge point set $S_{r2} = \{c_{nr2}, nr2 = 1, 2, \ldots, Nr2\}$

Ensure: Blob set $S_b^\star$

Compute the distance $D_{n1,n2}$ of each fork $(c_{n1}, l_{n1}) \in S_{f1}$ and $(c_{n2}, l_{n2}) \in S_{f2}$.
Get the nearest neighbor fork pair $\{(c_{n1}, l_{n1}), (c_{n2}, l_{n2}), D_{n1,n2}\}$ of the two sets.
if $D_{n1,n2} \leq \epsilon$ **then**
    add (c_{n1}, l_{n1}) into $S_{f1}^\star$ and (c_{n2}, l_{n2}) into $S_{f2}^\star$
else
    remove (c_{n1}, l_{n1}) if no closer neighbor fork in S_{f2}, do the same operation
    for (c_{n2}, l_{n2}).
end if
for each blob (c_m, σ_m, ν_m) **do**
    compute $D_m = |c_m - c_n'|$, where c_n' denotes the coordinate of either c_{n1}
    or c_{n2}, and it indicates an arbitrary fork points $(c_n', l_{n1}') \in S_{f1}^\star \bigcup S_{f2}^\star$.
    if $D_m = |c_m - c_n'| > \sigma_m$ **then**
        remove (c_m, σ_m, ν_m)
    else
        add (c_m, σ_m, ν_m) into S_b^{cf}
    end if
end for
for each blob $(c_i, \sigma_i, \nu_i) \in S_b^{cf}$ **do**
    compute the distance $D_{ij} = |c_i - c_j|$ with blob $(c_j, \sigma_j, \nu_j) \in S_b^{cf}$ where
    $i \neq j$
    if $max\{\sigma_i, \sigma_j\} < D_{ij} \leq (\sigma_i + \sigma_j)$ **then**
        add (c_i, σ_i, ν_i) into $S_b^\star$
    else if $D_{ij} \leq max\{\sigma_i, \sigma_j\}$ **then**
        count number n_i^c, n_j^c of $c_r \in S_{r1} \bigcup S_{r2}$ in the two blob regions and
        the point amount n_i^k, n_j^k of the two blob regions, respectively.
        if $(n_i^c/n_i^k) > (n_j^c/n_j^k)$ **then**
            add (c_i, σ_i, ν_i) into $S_b^\star$
        else if $(n_i^c/n_i^k) = (n_j^c/n_j^k)$ **then**
            if $\nu_i \geq \nu_j$ **then**
                add (c_i, σ_i, ν_i) into $S_b^\star$
            else
                remove (c_i, σ_i, ν_i)
            end if
        else
            remove (c_i, σ_i, ν_i)
        end if
    else
        add (c_i, σ_i, ν_i) into $S_b^\star$
    end if
end for

where β_i denotes the weight of the ith distance. Thus, the junction centers are the points which have the minimal value of the linear combination of three kinds of distances. Sometimes, a junction blob may not cover any J_b or J_c. For such cases, we extend the radius r of blobs to include the nearest J_b or J_c, and set a upper bound υ for r. In our experiment, $\upsilon = 8$ pixels. If there is still no J_c or J_b in the extended blob, we select the ridge line point which is the closest to a junction blob center, or the branch line point which is the closest to a junction blob center.

In addition to locating the line junction centers, we also estimate branch number and orientation. In a certain blob, since the junction center is very close to the branch converging point J_b, it is straightforward to get the branch number by counting the number of valid fork branches that intersect at J_b. Accordingly, branch orientations are estimated by the maximal filter orientation of the adjacent ridge lines. Considering that small blobs may contain no fork obtained by maximal filter response, we extent the radius of the blob to get the nearest fork, and then use the same procedure mentioned above.

IV. EXPERIMENTAL RESULTS AND DISCUSSION

A. EXPERIMENTAL SETTINGS AND EVALUATION CRITERIA

To evaluate the performance of JUNR, our experiments were performed on both synthetic and real images, including neuron and retinal vessel images. The synthetic data set is composed of simulated neuron images from NeuroMorpho.org [39],[1] and the real data set consists of retinal vessel images, i.e., the DRIVE data set [37].[2]

As for the synthetic data set, we selected 180 simulated neuronal SWC files randomly. Each file contains the morphology and log of a neuron which is presented as a sequence of linked center points as well as their corresponding radii. According to [1], the two dimensional fluorescence microscopy images were generated by the Gaussian point-spread function and Poisson noise. The line junction information for the simulated neuron images is contained in the raw files. For this kind of images, the preprocessing work is unnecessary due to the high quality of curvilinear structures.

The DRIVE data set is a widely used benchmark set for the evaluation of vessel segmentation algorithms. In our experiments, since our algorithm requires no labelled data, we did not differentiate training and test samples. And for evaluating our method, we used the ground truth data of junction center locations provided by Azzopardi and Petkov [18].[3] For images in DRIVE, we adopted the same preprocessing procedure described in [40], including homogenizing background intensity distribution and denoising. The main steps of the background homogenization and denoising are a serious of smoothing and shade-correction operations. To quantitatively analyze the performance of our approach, four quantities were recorded, namely true positive (TP), false positive (FP), true negative (TN) and false negative (FN). Specifically, TP represents the number of true junction centers detected, TN is the number of true background points that are not detected as junction center, FP is the number of falsely detected junction centers, and FN represents the number of true junction centers that are not detected. It is quite common to relax the criterion of accuracy, i.e., points that are within ρ pixels of a true junction center are also regarded as true positives. Here we set ρ to 5 as suggested in [16]. Several widely used statistical metrics were adopted, including sensitivity, specificity and their harmonic mean, which are calculated as $Precision = TP/(TP + FP)$, $Recall = TP/(TP + FN)$, and F_1 score $= 2 \times Precision \times Recall/(Precision + Recall)$.

In our method, the parameters mainly include weighting coefficients and parameters of filter kernel. The measurement coefficients in Eq. ((7)) are $\alpha_1 = 80$, $\alpha_2 = 4$, and $\alpha_3 = 1$ on all data sets, as suggested in [25]. The distance coefficients in Eq. ((14)) are $\beta_1 = \beta_2 = 5$, and $\beta_3 = 1$ for all of the 50 images. In the blob screening step, for simulated neuronal images, the window size is 9×9, and the stick length $2 \leq L \leq 7$; while for the DRIVE data set, the window size is 15×15, and the stick length $2 \leq L \leq 15$. The parameters of Gabor filter bank are set as follows: for simulated neuronal images, wavelength $\lambda = [2, 2\sqrt{2}, 4]$, aspect ratio is 0.5, bandwidth is 2, phase offset $\varphi = [0, \pi/2, \pi]$ and 24 orientations are $\{\theta = \frac{\pi}{18} i \mid i = 0, 1, 2, \ldots, 23\}$; for retinal vessel data set, wavelength $\lambda = [2, 2\sqrt{2}, 4, 4\sqrt{2}, 8]$, aspect ratio is 0.5, bandwidth is 2, phase offset $\varphi = [0, \pi]$ and 36 orientations are $\{\theta = \frac{\pi}{18} i \mid i = 0, 1, 2, \ldots, 35\}$.

B. EXPERIMENTAL RESULTS

In this section, we show both visual and statistical detection results on the two data sets. In the low level vision field, observers' judgement is a crucial evaluation criterion. Junction detection is a typical low level vision task, thus the visual results can directly reflect the quality of the detector, while the quantitative results make the performance evaluation more objective.

1) THE EXPERIMENT ON THE SYNTHETIC DATA SET

Fig. 6 shows several examples of the detection results of the simulated images, including *Drosophila melanogaster* and Chinese grey hamster neurons. We mark the junctions in that images, where red circles are the junction regions, green lines indicate the junction branches and yellow points mark the junction centers. Apparently, our method can locate the intersections accurately and represent their characteristics adequately.

For the assessment on simulated neuron images, we compared our method with an approach that is specialized for crucial point detection in neuron images proposed by Miroslav *et al.* [1], and a versatile method proposed by Su *et al.* [25]. The detection results are listed in Table 1. As can be seen, our method is superior to the other two methods on all the three criteria. Actually, all of the three

[1]The swc files can be downloaded at http://neuromorpho.org/
[2]http://www.isi.uu.nl/Research/Databases/DRIVE/.
[3]http://www.cs.rug.nl/imaging/databases/retinadatabase

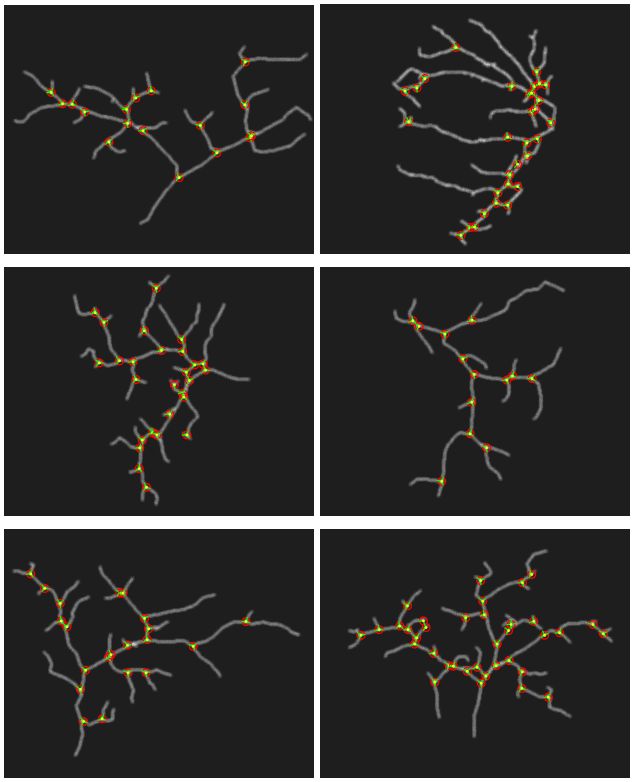


FIGURE 6. Experimental results on neuronal images (SNR = 5). Red circles are the detected junction regions, green lines indicate the branch orientations and yellow points denote the junction centers.

TABLE 1. Result comparison on the simulated neuron data set with $SNR = 5$.

Method	Precision (%)	Recall (%)	F_1 (%)
Radojevi's	84.52	85.73	85.13
Su's	88.75	90.12	89.43
JUNR	91.47	94.26	92.84

methods take advantage of the local neighborhood of points and compute the response of a directional filter bank. Radojevi's method utilizes a specifically designed fuzzy-logic rule system to estimate the junction. This method is good at termination detection, but is easily misguided by the local density intersections and tortile curvilinear structures. In Su's approach, the final junction position is confirmed by the intersection of its branch lines, and the branches are identified according to the similarities between adjacent directional sticks with their pre-defined templates. A disadvantage of this method is the difficulty in distinguishing nearby junctions when they gather closely. Our method could overcome the drawbacks of the two algorithms, as we select the junctions through a screening scheme, and locate the junction centers with the weight distances, which integrates three junction metrics.

Furthermore, we investigated the influence of signal-to-noise-ratio (*SNR*), and experimented with different SNR values, i.e., 2, 3, 4, 5, as suggested in [1]. JUNR performs better

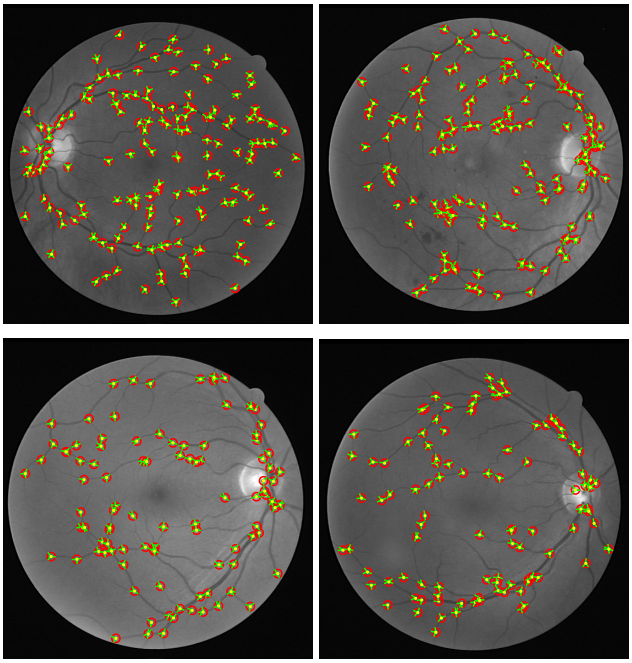


FIGURE 7. Experimental results on DRIVE images. The Red circles are the detected junction regions, green lines indicate the branch orientations and yellow points denote the junction centers.

than other two methods when $SNR \geq 3$; and when $SNR < 3$, its performance is slightly inferior to Su's but also superior to Radojevi's method. With the increase of noise ratio, the detection accuracies of all the three methods decreased. Our method reduces the impact of noise to some extend via the process of screening the blobs which may contain junction centers. And Su's approach is also partly overcome the impact of noise by locating the junction centers with local feature refinement, while Radojevi's method suffers more from noise because high noise can offset the feature values in the fuzzy logic rules.

2) THE EXPERIMENT ON THE DRIVE DATA SET

Fig. 7 shows JUNR's results on four DRIVE images. It can be observed that it achieves good performance on these images, but there are still some falsely detected points and missing points, which may be caused by two reasons. First, some puny curvilinear structures such as vessel twigs have been removed in the preprocessing step, thus the related line junctions may have low metric value close to that of the background points. Second, in the junction blob selection step, some overlapped blobs are removed. It is possible that the excluded blobs contain true junction center points. This will lead to loss or position deviation of junction center points.

We compared our method with several other approaches specialized for retinal vessel junction detection, including hybrid junction number method [2], filter based junction detector [30], Saha's method [31], GRAID (GRowing Algorithm for Intersection Detection) [16], and COSFIRE (Combination Of Shifted Filter Responses) [18]. All these

TABLE 2. Result comparison on the DRIVE data set.

Method	Precision (%)	Recall (%)	F_1 (%)
Aibinu's[a]	80.99	93.73	86.90
Modified Saha's[a]	85.70	91.79	88.64
Martinez-Perez's[a]	53.24	89.55	66.78
GRAID[a]	90.60	93.22	91.89
COSFIRE[a]	96.60	97.81	97.20
Su's [b]	86.13	91.92	89.36
JUNR[b]	91.76	94.32	93.02

[a] Prior segmentation is required. [b] Prior segmentation is not required.

methods detect junctions in binary images provided by manual segmentation. Aibinu *et al.* [2] thinned the binary images in which vessels are only one pixel wide, then they used a small window to count the connected pixels for determining the junction position and type. Filter based junction detector skeletonizes the binary images to get junction centers with no less than 3 neighbors from 8 adjacent pixels. Saha's method extracts junctions with 3 branches via counting the cyclic path of a sliding window with the radius of 3. GRAID does not thin the binary curvilinear structure, but computes the distance map of each pixel and gets a local distance cost to judge whether the point is a local ball center, then uses an expanded frontier function to infer the branch number and decide which local ball centers are junctions. COSFIRE algorithm calculates the blurred and shifted responses of selected Gabor filters to extract the junctions in the segmented vessel images. Besides, we compare with the Su's method (working on raw images as used in the DRIVE experiment). The detection performance is shown in Table 2. Our method ranks the second among the 7 methods. The first three algorithms need to skeletonize the binary images into one pixel wide line structures. This procedure produces bias in the thinning step and the final results rely on the window size. The detected junction location may have large shifts with the true junction centers. The GRAID method would result into high error rate when calculating the shortest branch. For example, if the binary image has a gap in vessel structure which is close to the junction region, the shortest branch judgement will be negative in the intersection model, and that junction center point will be excluded. The aforementioned disadvantage of Su's method also exists in dealing with real images, i.e., computing the directional stick response of local intensity is hard to distinguish the junctions close to each other. JUNR can overcome the these shortcomings to some extent since JUNR does not determine the junction in image directly but maps the images into the metric space, and restricts the potential junction centers within the junction blobs.

In addition, we also examine the distance between the detected position and true location of junction centers on DRIVE [37]. Intuitively, a small distance indicates good detection performance. Table 3 lists all the mean distances between correctly detected points with their corresponding ground truth, where all the mean distances are less than 3 pixels, which indicates that JUNR will still have a good assessment result with a stricter evaluation criteria.

TABLE 3. Results of JUNR on DRIVE images.

Image Index	GT	FP	FN	Mean Distance
1	135	10	8	0.7829
2	149	9	7	0.0118
3	143	13	10	2.3571
4	143	10	9	2.9610
5	156	11	8	1.8255
6	168	12	10	1.0783
7	147	10	8	2.8930
8	124	10	9	2.6197
9	134	9	6	1.3054
10	141	12	8	0.4598
11	182	14	9	1.9541
12	124	12	7	0.4815
13	168	13	7	0.9586
14	133	10	5	0.4465
15	84	8	6	0.4835
16	140	13	9	0.7563
17	112	11	7	1.4476
18	96	10	8	1.2843
19	116	12	6	2.6849
20	98	9	7	0.8627
21	107	11	5	0.9003
22	151	11	7	1.6944
23	48	5	5	2.3025
24	193	13	10	0.1193
25	122	11	6	1.0982
26	101	13	7	0.9178
27	133	12	8	1.8115
28	145	14	7	1.4676
29	153	15	9	0.7333
30	98	9	5	2.6172
31	65	6	5	1.5940
32	111	10	6	2.1950
33	118	9	5	1.2145
34	137	13	10	2.1660
35	127	11	6	2.1114
36	152	10	8	0.8921
37	133	11	7	0.2417
38	116	10	6	1.8575
39	114	13	9	1.4051
40	101	8	6	2.8585

The GT, FP and FN denote the junction number of ground truth, false positive and false negative, respectively. The last column shows the mean distance between the detected junction centers and true centers in the images, and the distance unit is pixel.

Note that the COSFIRE algorithm gets a better performance than all the other methods. COSFIRE obtains the optimal branches with Gabor filters by learning the parameters of the filters. Different from JUNR, COSFIRE works on binary images, and the learned filters have very different responses in curvilinear and background region. Although COSFIRE achieves very good performance and its detection process becomes much easier on binary images, the quality of labeling has great impact on its detection results. That is, the performance of COSFIRE method would fluctuate when the ground truth changes, as mentioned in [16]. In summary, JUNR achieves better or comparable performance to the existing state-of-the-art detectors of junctions. In many actual biomedical images, obtaining the curvilinear structure is a very hard work. In such cases, JUNR can be a versatile and powerful tool for the line junction detection task.

V. CONCLUSION

In this paper, we propose a new approach to detect and characterize line junctions in synthetic and real biomedical images. We first provide a formal definition of line junction and

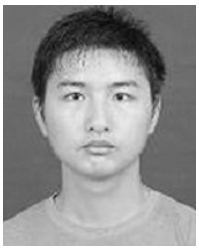
summarize several essential properties of it. Then we propose the JUNR method, which consists of four major steps, namely the generation of score matrix using a combined measurement, the detection of blobs containing junction centers, the screening of blobs, and the determination of junction center locations as well as branch characteristics. This new method achieves good performance on simulated neuron images and competitive results on real retinal vessel images compared against the state-of-the-art detectors for line junctions. The main advantage of JUNR is that the junction detection is based on the raw monochromatic images in which the line or tubular structure is unknown, and no labelled data is required. The JUNR is a flexible and versatile method for line junction detection in many biomedical images, especially when the images contain complex curvilinear networks which are hard to delineate.

ACKNOWLEDGEMENT

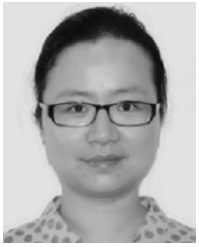
Hanjin Zhang and Yang Yang contributed equally to this work.

REFERENCES

[1] R. Miroslav, S. Ihor, and M. Erik, "Fuzzy-logic based detection and characterization of junctions and terminations in fluorescence microscopy images of neurons," *Neuroinformatics*, vol. 14, no. 2, p. 201, 2016.
[2] A. M. Aibinu, M. I. Iqbal, A. A. Shafie, M. J. E. Salami, and M. Nilsson, "Vascular intersection detection in retina fundus images using a new hybrid approach," *Comput. Biol. Med.*, vol. 40, no. 1, pp. 81–89, 2010.
[3] A. Fathi, A. R. Naghsh-Nilchi, and F. A. Mohammadi, "Automatic vessel network features quantification using local vessel pattern operator," *Comput. Biol. Med.*, vol. 43, no. 5, pp. 587–593, 2013.
[4] A. Can, H. Shen, J. N. Turner, H. L. Tanenbaum, and B. Roysam, "Rapid automated tracing and feature extraction from retinal fundus images using direct exploratory algorithms," *IEEE Trans. Inf. Technol. Biomed.*, vol. 3, no. 2, pp. 125–138, Jun. 1999.
[5] D. E. Sinzinger, "A model-based approach to junction detection using radial energy," *Pattern Recognit.*, vol. 41, no. 2, pp. 494–505, 2008.
[6] G.-S. Xia, J. Delon, and Y. Gousseau, "Accurate junction detection and characterization in natural images," *Int. J. Comput. Vis.*, vol. 106, no. 1, pp. 31–56, Jan. 2014.
[7] L. Parida, D. Geiger, and R. Hummel, "Junctions: Detection, classification, and reconstruction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 7, pp. 687–698, Jul. 1998.
[8] R. Elias and R. Laganiere, "JUDOCA: Junction detection operator based on circumferential anchors," *IEEE Trans. Image Process.*, vol. 21, no. 4, pp. 2109–2118, Apr. 2012.
[9] C. Harris, "A combined corner and edge detector," in *Proc. Alvey Vis. Conf.*, 1988, pp. 147–151.
[10] S. M. Smith and J. M. Brady, "SUSAN—A new approach to low level image processing," *Int. J. Comput. Vis.*, vol. 23, no. 23, pp. 45–78, 1997.
[11] E. Rosten and T. Drummond, "Fusing points and lines for high performance tracking," in *Proc. 10th IEEE Int. Conf. Comput. Vis. (ICCV)*, vol. 2. Oct. 2005, pp. 1508–1515.
[12] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *Proc. Eur. Conf. Comput. Vis.*, 2006, pp. 430–443.
[13] M. Maire, P. Arbeláez, C. Fowlkes, and J. Malik, "Using contours to detect and localize junctions in natural images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2008, pp. 1–8.
[14] Z. Zhang, S. H. Ong, X. Zhong, and W. C. K. Foong, "Efficient 3D dental identification via signed feature histogram and learning keypoint detection," *Pattern Recognit.*, vol. 60, pp. 189–204, Dec. 2016.
[15] M. Karpushin, G. Valenzise, and F. Dufaux, "Keypoint detection in RGBD images based on an anisotropic scale space," *IEEE Trans. Multimedia*, vol. 18, no. 9, pp. 1762–1771, Sep. 2016.
[16] J. M. Núñez, J. Bernal, F. J. Sánchez, and F. Vilariño, "Growing algorithm for intersection detection (GRAID) in branching patterns," *Mach. Vis. Appl.*, vol. 26, nos. 2–3, pp. 387–400, 2015.
[17] A. M. Aibinu, M. I. Iqbal, A. A. Shafie, M. J. E. Salami, and M. Nilsson, "Vascular intersection detection in retina fundus images using a new hybrid approach," *Comput. Biol. Med.*, vol. 40, no. 1, pp. 81–89, 2010.
[18] G. Azzopardi and N. Petkov, "Automatic detection of vascular bifurcations in segmented retinal images using trainable cosfire filters," *Pattern Recognit. Lett.*, vol. 34, no. 8, pp. 922–933, 2013.
[19] G. Azzopardi and N. Azzopardi, "Trainable COSFIRE filters for keypoint detection and pattern recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 2, pp. 490–503, Feb. 2013.
[20] S. Abbasi-Sureshjani, I. Smit-Ockeloen, E. Bekkers, B. Dashtbozorg, and B. T. H. Romeny, "Automatic detection of vascular bifurcations and crossings in retinal images using orientation scores," in *Proc. IEEE Int. Symp. Biomed. Imag.*, 2016, pp. 189–192.
[21] F. Deschênes and D. Ziou, "Detection of line junctions and line terminations using curvilinear features," *Pattern Recognit. Lett.*, vol. 21, no. 6, pp. 637–649, 2000.
[22] F. Deschenes, D. Ziou, and M.-F. Auclair-Fortier, "Detection of lines, line junctions and line terminations," *Int. J. Remote Sens.*, vol. 25, no. 3, pp. 511–535, 2004.
[23] G. Azzopardi and N. Petkov, "Detection of retinal vascular bifurcations by trainable V4-like filters," in *Computer Analysis of Images and Patterns*. Berlin, Germany: Springer-Verlag, 2011, pp. 451–459.
[24] J. Chen, Y. Sato, and S. Tamura, "Orientation space filtering for multiple orientation line segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 5, pp. 417–429, May 2000.
[25] R. Su, C. Sun, and T. D. Pham, "Junction detection for linear structures based on hessian, correlation and shape information," *Pattern Recognit.*, vol. 45, no. 10, pp. 3695–3706, 2012.
[26] R. Su, C. Sun, C. Zhang, and T. D. Pham, "A new method for linear feature and junction enhancement in 2D images based on morphological operation, oriented anisotropic Gaussian function and hessian information," *Pattern Recognit.*, vol. 47, no. 10, pp. 3193–3208, 2014.
[27] T. A. Pham, M. Delalandre, S. Barrat, and J. Y. Ramel, "Accurate junction detection and characterization in line-drawing images," *Pattern Recognit.*, vol. 47, no. 1, pp. 282–295, 2014.
[28] S. He, M. Wiering, and L. Schomaker, "Junction detection in handwritten documents and its application to writer identification," *Pattern Recognit.*, vol. 48, no. 12, pp. 4036–4048, 2015.
[29] D. Calvo, M. Ortega, M. G. Penedo, and J. Rouco, "Automatic detection and characterisation of retinal vessel tree bifurcations and crossovers in eye fundus images," *Comput. Methods Programs Biomed.*, vol. 103, no. 1, pp. 28–38, 2011.
[30] M. E. Martinez-Perez *et al.*, "Retinal vascular tree morphology: A semi-automatic quantification," *IEEE Trans. Biomed. Eng.*, vol. 49, no. 8, pp. 912–917, Aug. 2002.
[31] S. Saha and N. D. Roy, "Automatic detection of bifurcation points in retinal fundus images," *Latest Res. Sci. Technol. Int. J.*, vol. 2, no. 2, pp. 105–108, 2013.
[32] C. Steger, "An unbiased detector of curvilinear structures," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 2, pp. 113–125, Feb. 1998.
[33] C. Steger, "Unbiased extraction of lines with parabolic and Gaussian profiles," *Comput. Vis. Image Understand.*, vol. 117, no. 2, pp. 97–112, 2013.
[34] M. W. Law and A. C. Chung, "Three dimensional curvilinear structure detection using optimally oriented flux," in *Proc. Comput. Vis.—ECCV*, 2008, pp. 368–382.
[35] J. Shi and C. Tomasi, "Good features to track," in *Proc. Comput. Vis. Pattern Recognit.*, Jun. 1994, pp. 593–600.
[36] T. Lindeberg, "Feature detection with automatic scale selection," *Int. J. Comput. Vis.*, vol. 30, no. 2, pp. 79–116, 1998.
[37] J. J. Staal, M. D. Abramoff, M. Niemeijer, M. A. Viergever, and B. van Ginneken, "Ridge-based vessel segmentation in color images of the retina," *IEEE Trans. Med. Imag.*, vol. 23, no. 4, pp. 501–509, Apr. 2004.
[38] U. T. Nguyen, A. Bhuiyan, L. A. Park, and K. Ramamohanarao, "An effective retinal blood vessel segmentation method using multi-scale line detection," *Pattern Recognit.*, vol. 46, no. 3, pp. 703–715, 2013.
[39] G. A. Ascoli, D. E. Donohue, and M. Halavi, "Neuromorpho.org: A central resource for neuronal morphologies," *J. Neurosci.*, vol. 27, no. 35, pp. 9247–9251, 2007.
[40] D. Marin, A. Aquino, M. E. Gegúndez-Arias, and J. M. Bravo, "A new supervised method for blood vessel segmentation in retinal images by using gray-level and moment invariants-based features," *IEEE Trans. Med. Imag.*, vol. 30, no. 1, pp. 58–146, Jan. 2011.



HANJIN ZHANG received the M.S. degree in control science and engineering from the University of Shanghai for Science and Technology in 2012. He is currently pursuing the Ph.D. degree with the Institute of Image Processing and Pattern Recognition, Shanghai Jiao Tong University. His research interests include image analysis, and object detection and recognition.



YANG YANG received the Ph.D. degree in computer science from Shanghai Jiao Tong University, China, in 2009. She was a Visiting Scholar with the University of California at Riverside from 2012 to 2013. She is currently an Assistant Professor with the Department of Computer Science, Shanghai Jiao Tong University. Her research interests focus on machine learning and bioinformatics.

HONGBIN SHEN received the Ph.D. degree from Shanghai Jiao Tong University, China, in 2007. He was a Post-Doctoral Research Fellow of the Harvard Medical School from 2007 to 2008. He was a Visiting Professor with the University of Michigan in 2012. He is currently a Professor with the Institute of Image Processing and Pattern Recognition, Shanghai Jiao Tong University. His research interests include pattern recognition and bioinformatics. He has published over 100 journal papers and constructed 35 bioinformatics severs in these areas. He has served as an Associate Editor of BMC Bioinformatics. Prof. Shen is an editorial board member of several international journals. He was an ESI Highly Cited Researcher from 2014 to 2015.